BIOGAN: Fingerprint Domain Adaption

MTP Report submitted to

Indian Institute of Technology Mandi

for the award of the degree

of

B. Tech

by

Raghav Sethi

under the guidance of

Dr Aditya Nigam

SCHOOL OF COMPUTING AND ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY MANDI

June 2018

CERTIFICATE OF APPROVAL

Certified that the thesis entitled **BIOGAN: Fingerprint Domain Adaption**, submitted by **Raghav Sethi** , to the Indian Institute of Technology Mandi, for the award of the degree of **B. Tech** has been accepted after examination held today.

Date : 28th May, 2018
Mandi, 175001

Faculty Advisor: Dr Padmanabhan Rajan

CERTIFICATE

This is to certify that the thesis titled **BIOGAN: Fingerprint Domain Adaption**, submitted by **Raghav Sethi**, to the Indian Institute of Technology, Mandi, is a record of bonafide work under my (our) supervision and is worthy of consideration for the award of the degree of **B. Tech** of the Institute.

Date : 28th May, 2018
Mandi, 175001

Supervisor: Dr Aditya Nigam

DECLARATION BY THE STUDENT

This is to certify that the thesis titled **BIOGAN: Fingerprint Domain Adaption**, submitted by me to the Indian Institute of Technology Mandi for the award of the degree of **B. Tech** is a bonafide record of work carried out by me under the supervision of **Dr Aditya Nigam**. The contents of this MTP, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Date : 28th May, 2018
Mandi, 175001

Raghav Sethi

Acknowledgments

I would like to express my deepest gratitude and special thanks to Dr Aditya Nigam for his guidance and supervision, who in spite of being extraordinarily busy with his duties, took time out to hear, guide and keep me on the correct path and motivated me at every step during the project.

I express my deepest thanks to Ms Avantika and Mr Daksh Thapar for taking part in weekly presentations and discussions, and giving necessary advices and guidance. I choose this moment to acknowledge their support gratefully. I would like to thank my classmates for their wonderful collaboration. You supported me greatly and were always willing to help me. At last, I would like to thank my college Indian Institute of Technology Mandi for giving me the opportunity to work on this project.

Raghav Sethi

Abstract

Biometric authentication has been broadly applied in computer security, law enforcement, banking, etc since the 2000s. However, the biometric recognition algorithms claim to have high performance but the performance is not good to enough to work practically in each and every case. The algorithms sometimes require a huge amount of training data. In particular, fingerprint sensors sometimes fail to recognise the person or take time to judge whether the person is authorised or not. It may be the problem of sensor lens or it may have been the algorithm it is trained on. We propose an approach to solve these problems using Generative Adversarial Networks(GAN) which are considered to be the best generative model in terms of image generation and style transfer. Our aim is to determine a mapping between two fingerprint domains such that low-quality fingerprint images could be converted to high-quality ones of the same subject which could be done by CycleGAN. We propose using Siamese loss in addition to the cyclic loss in the CycleGAN to improve the results. Qualitative results are presented for different GAN models and different datasets. Quantitative results of fingerprint matching using trained Siamese Network and Minutiae Cylinder code demonstrate the superiority of our approach.

Keywords: *Generative Adversarial Network, CycleGAN, Siamese Network, fingerprint matching*

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The world around us is a three-dimensional environment containing enormous amount of information. Generative models are an essential part of Artificial Intelligence which can help us to understand and analyse data. They are a powerful unsupervised learning technique and have achieved immense success in the recent years. We train a generative model by providing it large amount of data of some domain. They aim to learn true data distribution of training data so as to produce new data points with some variety.

These days from smartphones to hostel attendance systems different fields are using biometric authentication. With the advent of new technology, the computation power is increasing exponentially which computer scientists take advantage of and develop faster algorithms every year. The biometric systems are not perfect and vulnerabilities in them have been exploited in the past. They have been vulenrable to presentation attacks - vulnerable to someone using a biometric data to match as someone else.

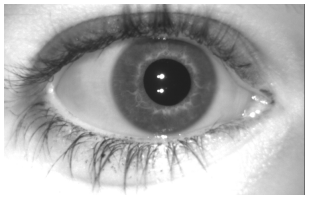


Fig. 1.1: An image from the Iris Dataset

In this project we propose a neural-network based method using Generative Adversarial Networks for synthesizing biometric data with the two purposes - to transform fingerprints

from one domain to another and to provide the industry with a large biometric dataset to test biometric recognition algorithms and improving the security level of biometric systems.

1.1 Motivation

We know that personal authentication is an essential social requirement. From smartphones to attendance systems, personal authentication is implemented everywhere. The authentication systems which have been developed previously are either token or knowledge based, hence they only provide limited protection. Biometrics can be considered as an option as they are easy to use and hard to deceive.

Biometric security systems were proposed long back. However, it has got the attention of researchers lately. New biometric encoding and processing algorithms have been developed but haven't gone considerate testing due to lack of availabilty of datasets. This has lead to not-so-secure security systems which could be bypassed by various attacks using synthetic biometric data.

Fingerprints and iris scans are commonly, and increasingly, used for authentication in a large variety of systems, from doors to workstations and smartphones. But they are potentially vulnerable to presentation attacks. Due to the demand for quick matches, low cost, and the decreasing space left for sensors in mobile system, presentation attacks are potentially practical, if a method for generating useful synthetic biometric data can be found.

Sometimes we observe that our fingerprints don't authenticate on the first time when we try a new scanner. This is because fingerprint recognition algorithms are built for a particular sensor. If we recorded our fingerprint on a low cost sensor, it may save less information or different information in the database with some noise and when we scan our finger on a high cost sensor it won't match to the fingerprint stored in the database because the features extracted don't match. It is possible that some intruder can may benefit from this flaw.

In crime scenes, the police force try to find any fingerprints. The fingerprints at the crime scenes are the type of latent fingerprints which can be extracted using magenetic powder. These fingerprints are difficult to match. If we could somehow convert them to different form the problem could be solved.

1.2 Objective

The aim of the project is to achieve Inter-Sensor Fingerprint Domain Adaption using Generative Adversarial Network(GAN). We would also generate huge quantity of high quality synthetic biometric images. The images could be used as a biometric dataset by the research community to test newly-designed algorithms and to improve biometric systems. The dataset could also be used in deep hashing algorithms. A proper analysis would be performed by caluclating matching scores of real and generated images to understand the efficacy of the proposed approach. The fingerprint analysis would be done by two methods - firstly, finding similar minutiae and calculating similarity and secondly by calculating scores from a Siamese network trained on fingerprints.

Chapter 2

Background and Related Work

2.1 Generating Synthetic Images

2.1.1 Earlier Approaches

Synthetic iris images were first generated by Cui et al [2]. It served the purpose to increase the dataset for iris recognition algorithms. He used Principal Component Analysis(PCA) and other techniques to improve resolution to create new iris images. There has been a model developed by Zuo et al. [3] based on anatomy to produce iris images. Ross and Shah [4] exploited Markov Random Field to make the texture and features of fake iris images. Galbally et al. [1] has reproduced iris from the feature template. They also did successful matching to real iris images. Figure 2.1 shows some produced fake iris images from Synthetic DataBase by Galbally et al. [1]. However, we can observe that the fake iris do not follow the real iris and turn out as fraudulent.

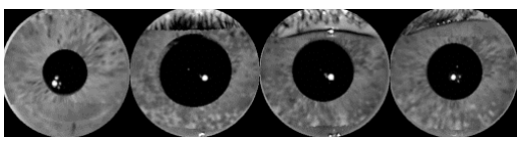


Fig. 2.1: Sample images from Synthetic DataBase [1]

2.1.2 Using Neural Networks

There has been rapid advancements in image generation by way of neural networks. Some of the most popular methods for image generation are Fully Visible Belief Networks (FVBN), Variational Autoencoders (VAE), and Generative Adversarial Networks (GAN). FVBNs such as PixelRNN produce one pixel at a time, similar to text generation and can have a bit of noise in their output. VAEs on the other hand tend to produce very smooth outputs. VAEs are used to generate images having specific features. Current GAN methods are perceived to produce the best results with fewer artifacts than FVBNs and sharper images than VAEs

2.1.3 Generative Adversarial Networks

GANs [5] have achieved remarkable results in image generation. They consist of a discriminator and a generator. The generator's task is producing fake images and the discriminator's task is to give a score to an image whether it's real or not. They are trained together simultaneously trying to out-perform each other. A lot of variety of GANs have been published till now.

2.2 Fingerprint Spoofing

Roy et al. [6] studied the vulnerability of fingerprint-based biometric systems that have small sensors for authentication and therefore only scan part of the fingerprint. They found these systems are highly susceptible to a type of presentation attack that is known as a wolf attack. A "wolf" is a biometric sample, real or synthesized, that impersonates multiple subjects biometrics. Roy et al. [6] showed that there exists synthetic fingerprints that can match for many fingerprints in a dataset. Their method represents fingerprints as minutiae templates. Many fingerprint identification systems will first identify the minutiae in the fingerprint and then compare them to the minutia template saved in the system.

With the help of GANs we would generate images directly instead of working at the minutia level which is better as it would generate actual synthetic images which would also

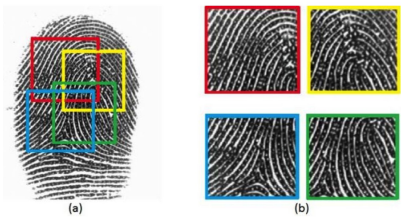


Fig. 2.2: (a) A complete fingerprint (b) parts of a fingerprint which could be scanned by the sensor

work for the systems which don't use minutiae templates.

2.3 Image-to-Image translation

Image-to-image translation was first thought and implemented by Hertzmann et al.s Image Analogies [7]. Non-parametric texture model [9] was used on a single input-output training image pair. Pix2Pix implemented by Isola et al. [8] used a Conditional GAN to learn a mapping from input to output images. Using related approaches people have developed models to produce photos from sketches and coloured photos from black and white photos [9]

Chapter 3

Model Implementation and the Work done

3.1 Generative Adversarial Networks

GANs are a impicit type of generative models that learn to generate images in a semi-supervised fashion. There are two parts to GAN; a generator and a discriminator. Figure 3.1 is the basic model of a GAN. The generator is a neural network that takes random noise as an input and outputs an image. The discriminator is also typically a neural network, it takes an image as an input and classifies it as real or generated. We studied some of the GANs, found which would work on our biometric data and finally experimented with them. The GANs we studied were DCGAN, CGAN, BEGAN,Pix2pix and CycleGAN.

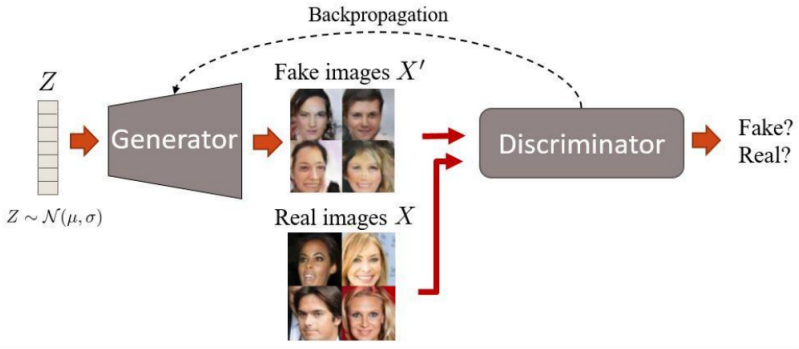


Fig. 3.1: GAN Model

GAN objective:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \tag{3.1}$$

Here:

- z: Random noise
- $G(z)$: Generated Image
- x: Real Image
- $D(x)$: Discriminator Score on Real Image
- $D(G(z))$: Discriminator Score on Generated Image

The generator and the discriminator a play a game of min-max with each other. The generator tries to maximise $V(D,G)$ and the discriminator tries to minimise it.

We had to adjust hyperparameters, network architecture and training procedure for different datasets. After shifting from CelebA dataset, things were adjusted to incorporate different type of data. The GANs gave better results in case of fingerprints as compared to the iris dataset. We worked on DCGAN and BEGAN on a lot of datasets and produced huge quantity of biometric images.

3.2 CycleGAN

CycleGAN learns to translate an image from a source domain X to a target domain Y in the absence of paired examples [10].We use two different fingerprint sensors data as two different domains and try to learn the mapping between them. It consists of two GANs coupled together to learn mapping between the domains. It contains cyclic loss as well as adversarial loss.

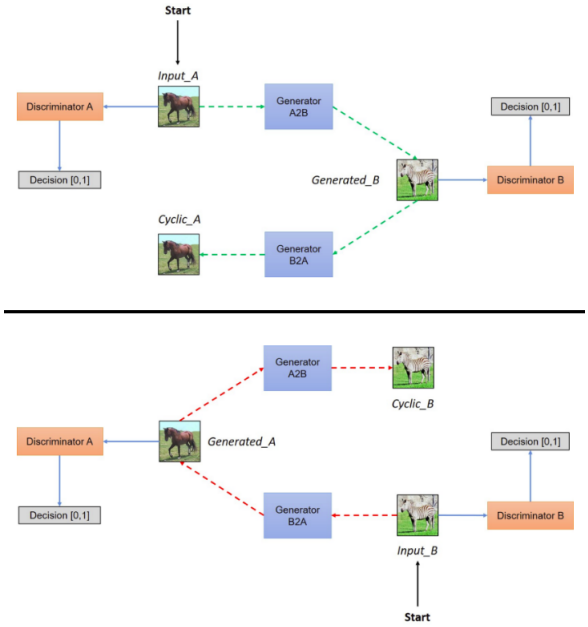


Fig. 3.2: CycleGAN Model

The generator of the CycleGAN consists of three parts- encoder, transformer and decoder. Earlier, we were using UNet model in Generator but the generator was failing to produce the fingerprints. So after that we implemented the ResNet Model which consists of 150 layers.

We have many datasets of fingerprints from different sensors. However their sizes are different and also quality of the fingerprint varies a lot from sensor to sensor. Pix2Pix is a Conditional Generative Adversarial Network which is used to generate images from one domain to other with the help of paired-datasets of the two image domains. If we could run Pix2Pix on fingerprints of two different sensors we could learn the mapping of how to convert one type of fingerprint to the other type. However, currently we don't have fingerprint data of same subject from two different sensors.This problem could be solved by CycleGAN.

The CycleGAN could convert a fingerprint from a low cost sensor to that of a high quality sensor. So for collecting data which happens at many places we could use cheap sensors and convert the fingerprint to that of high quality sensor and store in the database. So that when the fingerprint is verified from a high-quality sensor at some different place it matches the fingerprint present in the database. The CycleGAN could also convert a latent

fingerprint to a different type which makes matching possible.

We used three different types of fingerprints for domain transfer to be learned by the CycleGAN. We segregated the data into training and testing. The testing data was used in fingerprint matching algorithms.

3.3 Siamese Network

Siamese neural network has the objective to find how similar two comparable things are [11]. It contain two or more identical sister networks. The sister networks have the same configuration with the same parameters and weights. Parameter updating is mirrored across both sister networks.

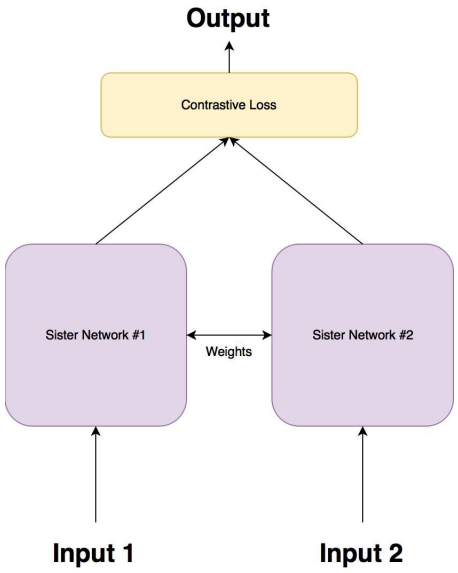


Fig. 3.3: Siamese Network Model

We trained a Siamese Network on Lumidigm and Futronics fingerprint images using two types of losses- contrastive loss and triplet loss. Triplet loss is known to be a better method to train a Siamese Network and it takes a lot of time to train also. During training, we take a batch of three images. We set one of them as an anchor and set the others as positive or negative depending upon the similarity. The similarity between the anchor and positive can be lesser than the similarity between the anchor and the negative must be low. What we do during training is we try to bring the anchor and positive image closer and anchor and

negative farther in the embedding space.

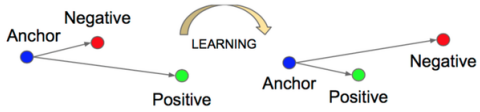


Fig. 3.4: Triplet loss training

3.4 Siamese-CycleGAN

The CycleGAN consist of cyclic loss in addition adversarial loss. What we propose is adding a trained Siamese network to the generators to incorporate an additional Siamese loss. Each generator is now trained using adversarial loss from the gradients of the discriminator, cyclic loss from the reconstructed image produced by the other generator, and the Siamese score from the trained Siamese network.

The code for merging the CycleGAN and Siamese network was written in Tensorflow Slim API. We created inference function which we would call to get the embeddings of an image we want. We then calculate the Siamese score which is the eucledian distance between the images. The score is added to the generator loss. While training the GAN we don't update the weights of the Siamese Network.

Chapter 4

Exerimental Studies and Results

Generative Adversarials Networks were proposed in 2014 and since then a lot of work has been done upon them. There have been a variety of GANs published which follow the same basic principal of adversarial loss. We studied some of the GANs, found which would work on our biometric data and finally experimented with them. The GANs we studied were DCGAN, CGAN, BEGAN and CycleGAN,

4.1 DCGAN on CelebA Dataset

We implemented DCGAN in Keras, Python and wanted to check whether it produced any fruitful results. So we took a simple dataset(CelebA) to test the code.

- Training Data: 20,000 images
- Optimiser: Stochastic Gradient Descent
- Learning Rate: 0.0005 momentum: 0.9
- Loss: Binary Cross-entropy

The training data had positions of eyes at same position in each image. DCGAN also learned to produce descent images having eyes at same position.

Fig. 4.1: Images produced by DCGAN implementation on CelebA dataset

4.2 DCGAN on Secugen Dataset

After CelebA datset we worked upon our biometric data. Secugen dataset consists of low quality fingerprints obtained from a low cost sensor.

- Training Data: 10,000 images
- Optimiser: Adam
- Learning Rate:0.00001, beta1=0.9, beta2=0.999
- Loss: Binary Cross-entropy

Fig. 4.2: Training dataset(on top), Images produced by DCGAN implementation on fingerprint dataset(on bottom)

The GAN failed to create ridges in the fingerprints(Figure 4.2). The generator only learned that there is some black portion in the middle and the rest is white.

4.3 DCGAN on Lumidigm Dataset

Lumidigm dataset consist of high quality fingerprint images taken from a high-cost sensor. Unlike Secugen dataset, the fingerprints were of same shape and size.

- Training Data: 14,000 images
- Optimiser: Stochastic Gradient Descent
- Learning Rate:0.00001, beta1=0.9, beta2=0.999
- Loss: Binary Cross-entropy

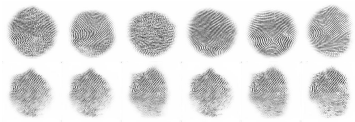


Fig. 4.3: Training dataset(on top), Images produced by DCGAN implementation on fingerprint dataset(on bottom)

The DCGAN performed good and produced ridges in the fingerprints(Figure 4.3). After certain iterations, the DCGAN suffered from mode collapse and started producing similar images. To check the uniqueness, features were extracted from each generated fingerprint and matched with the whole training dataset to find the closest matching real fingerprint using NFIQ code which is used for fingerprint analysis. The results were stored in text files.

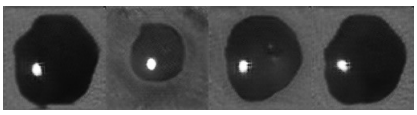


Fig. 4.4: Training dataset(on top), Images produced by DCGAN implementation on Lum fingerprint dataset(on bottom)

We used NIST Biometric Image Software to match fingerprints. First minutiae of both real and fake fingerprints were extracted. Fingerprints are matched upon the score given by no. of minutiae found same in both the fingerprints.

- On an average 50 minutiae are extracted from both real and fake fingerprints

- When fake fingerprints were matched with real ones, an average score of 40 was obtained.
- When fake fingerprints were matched within themselves the score ranged from 4 to 490.

4.4 DCGAN on Iris dataset

Iris dataset is available in two forms. The first dataset consist of grayscale images of iris. The first contain around 10,000 images of iris and the second dataset consists of 784 images is in the form of iris strips containing radial information of the iris.

- Training Data: 10,000 images
- Optimiser: Stochastic Gradient Descent
- Learning Rate:0.00001, beta1=0.9, beta2=0.999
- Loss: Binary Cross-entropy

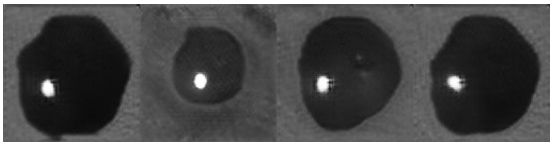


Fig. 4.5: Images produced by DCGAN implementation on iris dataset

The DCGAN didn't work well in the case of iris images(Figure 4.5). The reason might be that the training dataset didn't have fixed position of pupil in the images and we have cropped the images approximately at the centre.

The second dataset consist of iris strip images. The strip contains radial information of the iris. They seem to have complex patterns to naked eyes.

- Training Data: 784 images
- Optimiser: Stochastic Gradient Descent

- Learning Rate:0.00001, beta1=0.9, beta2=0.999
- Loss: Binary Cross-entropy

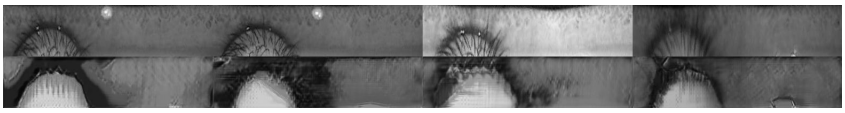


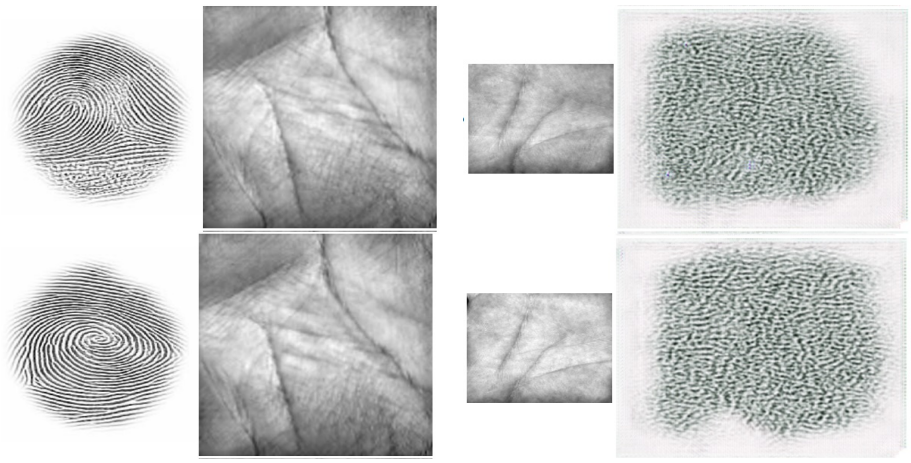
Fig. 4.6: Training dataset(on top), Images produced by DCGAN implementation on iris strip dataset(on bottom)

The iris strips produced blurred images having blocks in the images. The DCGAN failed to learn the intricate features of the iris.

4.5 CycleGAN on Lumidigm and Palm Dataset

We implemented CycleGAN on Tensorflow, Python. We had chosen two different domains to learn domain tranfer.

- Training Data: 7600 images
- Optimiser: Adam
- Learning Rate:0.0002, beta1=0.5
- Epochs: 20



(a) Figure A (b) Figure B

Fig. 4.7: Fig A: finger2palm, Fig B:palm2finger

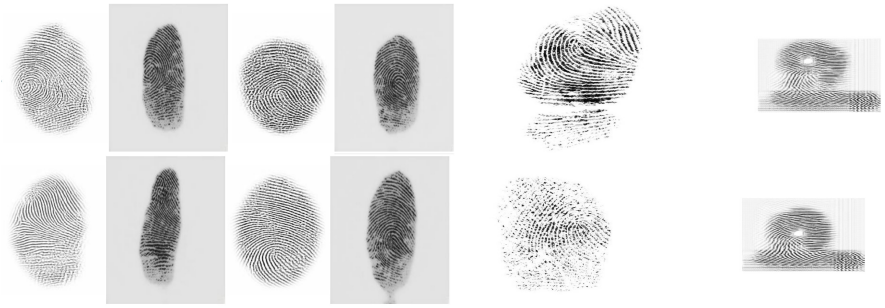
The code took 3 days to run. The CycleGAN learned to make palm prints from fingerprints but the reverse was not that good(Figure 4.7). It could be because the fingerprint has much intricate pattern which is difficult to learn which is not possible to be learned from the less complex palm prints.

4.6 CycleGAN on Lumidgm and Secugen Dataset

The Secugen has poor quality of fingerprints as compared to Lumidigm Dataset.

- Training Data: 5000 images
- Optimiser: Adam
- Learning Rate:0.0002, beta1=0.5
- Epochs: 115

The code took 4 days to run. The CycleGAN learned to make high-quality Secugen fingerprints from the Lumidigm fingerprint(Figure 4.8a). The fingerprint produced seems that it belonged to same subject from which the Lumidigm fingerprint was obtained. The CycleGAN learned to map the points from where ridges were curling. However, it failed to



(a) Figure A (b) Figure B

Fig. 4.8: Fig A: Lumidigm2Secugen, Fig B:Secugen2Lumidigm

learn the reverse mapping(Figure 4.8b). It could be because the Secugen fingerprints are of poor quality and could not be useful for the CycleGAN to learn the complex features of the Lumidigm fingerprint.

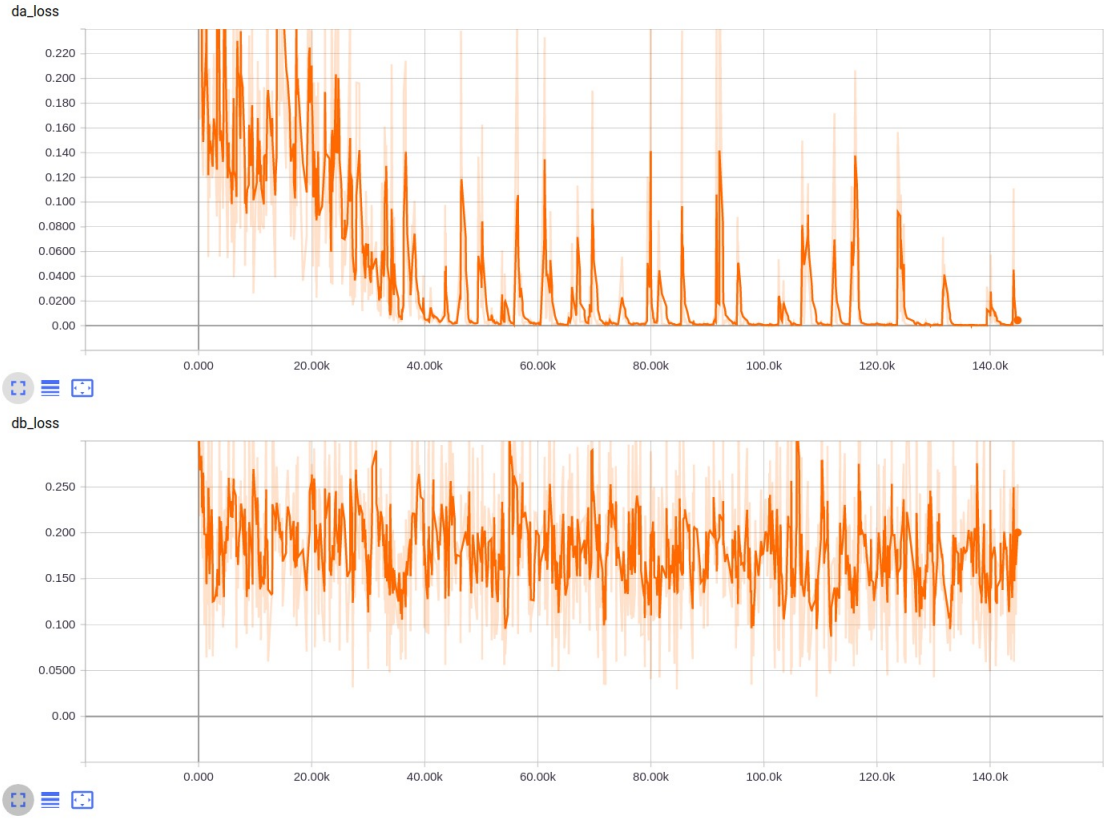


Fig. 4.9: Graphs showing Discrinator loss (on top: Lumidigm, on bottom: Secugen) with course of training. On Y-axis: Discriminator loss value, on X-axis: no. of steps in training

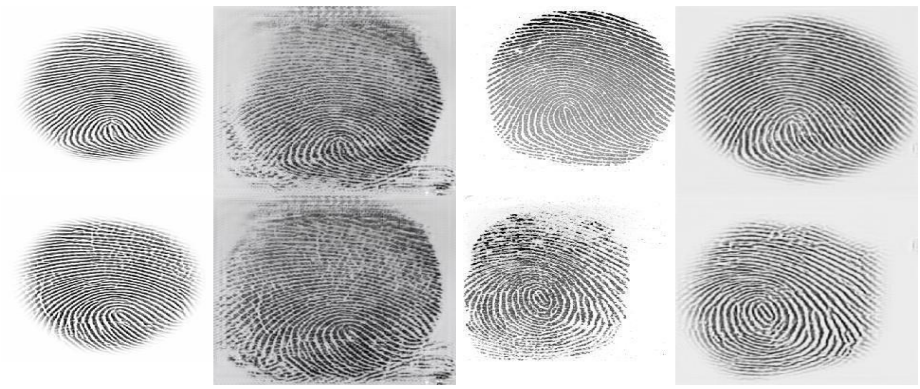
The graphs(Figure 4.9) were taken in real-time with the help of TensorBoard. It shows

that loss Discriminator of Lumidigm fingerprint decreases with training as it easily discriminates the real and fake because of poor performance of Generator producing fake Lumidigm fingerprints. However from the graph of Discriminator of Secugen we observe that the loss doesn't decrease but insead fluctuates. We can infer that the discriminator faces difficulty in discriminating real and fake Secugen prints.

4.7 CycleGAN on Lumidgm and Futronics Dataset

The Futronics dataset has better quality fingerprints than Secugen but not than Lumidigm.

- Training Data: 7600 images
- Optimiser: Adam
- Learning Rate:0.0002, beta1=0.5



(a) Figure A (b) Figure B

Fig. 4.10: Fig A: Lumidigm2Futronics, Fig B:Futronics2Lumidigm

The CycleGAN took a lot of time to produce good images. It recognised the circular shape of Lumidigm fingerprints. In early iterations the fingerprints were having a dot of white colour which it learned from some training image, suffered from mode collapse and it produced in the test images.

4.8 Siamese-CycleGAN on Lumidgm and Futronics Dataset

We modified the CycleGAN to incoporate a Siamese Network.

- Training Data: 7600 images
- Optimiser: Adam
- Learning Rate:0.0002, beta1=0.5

(a) Figure A (b) Figure B

Fig. 4.11: Fig A: Lumidigm2Futronics, Fig B:Futronics2Lumidigm

We ran the code for few epochs and the results were quite good as compared to the previous experiments. The Siamese loss helped the CycleGAN to learn to make better Lumidigm fingerprint images.

4.9 Biometric Evaluation of fingerprints produced

The fingerprints produced in the Section 4.7 and 4.8 were analysed using two methods-using trained Siamese network to calculate scores and using Minutiae Cylinder Code to find similarity. We generated text files comparing different fingerprints:

- Lumidigm Real v Lumidigm Real
- Lumidigm Real v Futronics Real

- Lumidigm Real v Lumidigm Generated

For each text file the records were stored in following manner:

Table 4.1: Results of Fingerprint Matching

Subject 1	Pose 1	Subject 2	Pose 2	Genuine/Imposter	Score

Each subject had 6 poses. We labelled '1' for genuine(meaning same subjects) and zero for imposter(meaning different subjects). The results were analysed by setting a threshold score and value and measuring the percentage of correct fingerprint matching. It was found that incorporting Siamese loss had better performance.

- False Acceptance Rate(FAR): Proportion of impostor attempts that are falsely declared to match a template of another subject.
- False Rejection Rate(FRR): Proportion of genuine attempts that are falsely declared not to match a template of the same subject.
- Correct Recognition Rate(CRR): Proportion of correctly matched subjects.

We calculated fingerprint matching scores using two methods:

- Minutiae Cylinder Code

We extracted minutiae from the real Lumidigm fingerprints and created a database. The other fingerprints were matched with the help of this database.

- Trained Siamsese Network

We trained a Siamese Network on real Lumidigm fingerprints and used it to calculate similarity score between two fingerprints.

Table 4.2: CRR on different matchings

	lumR_futR	lumCG_lumR	lumSCG_lumR
MCC	52.77	7.73	37.7
Siamese	2.7	10	10.5

From the table we can observe that Siamese-CycleGAN had better performance than the CycleGAN. In case of MCC matcher we had lower CRR the Futronics dataset but in case of Siamese Checker, Siamese-CycleGAN had the best performance.

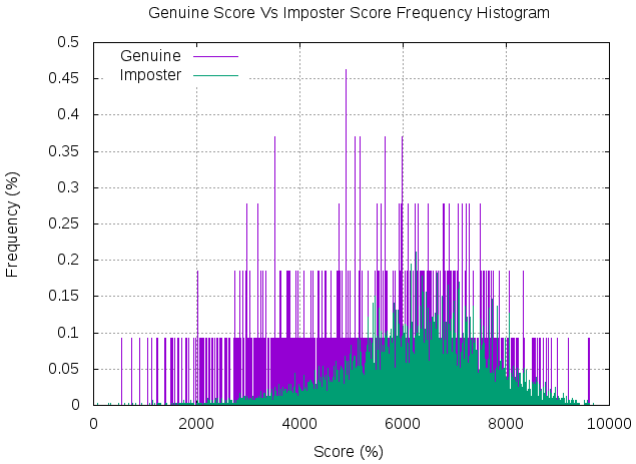


Fig. 4.12: Genuine Score vs Imposter Score frequency histogram on Lumidigm dataset, Siamese-CycleGAN

The two distributions should be separate enough so that a threshold could be set to identify genuine and imposter attempts. A good threshold minimises FAR and FRR. All the histograms generated on different datasets were roughly similar.

Chapter 5

Conclusion and Future Work

Conclusion:

After gathering and analysing the results from all the experimentation, we infer:

- DCGAN could be used to create fake fingerprints having unique minutiae.
- The quality of the fingerprints depend upon the quality of the training data.
- We require better training data is required for iris images so as DCGAN can learn properly.
- Fingerprint Domain Adaption is possible with the help of CycleGAN
- Secugen fingerprints produced by the CycleGAN were better than even the training dataset.
- Lumidigm fingerprints produced from Futronics were better as compared to the ones produced from the Secugen fingerprints
- Adding Siamese loss to the CycleGAN improved its performance.

Future Work:

- Changing the network models for different image sizes of fingerprints
- Training Siamese Network on other fingerprint domains
- Training Siamese-CycleGAN on other fingerprint domains

References

[1] J. Galbally, A. Ross, M. Gomez-Barrero, J. Fierrez, and J. Ortega-Garcia, "Iris image reconstruction from binary templates: An efficient probabilistic approach based on genetic algorithms," *Computer Vision and Image Understanding*, vol. 117, no. 10, pp. 1512–1525, 2013.

[2] J. Cui, Y. Wang, J. Huang, T. Tan, and Z. Sun, "An iris image synthesis method based on pca and super-resolution," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 4. IEEE, 2004, pp. 471–474.

[3] J. Zuo, N. A. Schmid, and X. Chen, "On generation and analysis of synthetic iris images," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 1, pp. 77–90, 2007.

[4] S. Shah and A. Ross, "Generating synthetic irises by feature agglomeration," in *Image Processing, 2006 IEEE International Conference on*. IEEE, 2006, pp. 317–320.

[5] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'14. Cambridge, MA, USA: MIT Press, 2014, pp. 2672–2680. [Online]. Available: http://dl.acm.org/citation.cfm?id=2969033.2969125

[6] A. Roy, N. Memon, and A. Ross, "Masterprint: Exploring the vulnerability of partial fingerprint-based authentication systems," *IEEE Transactions on Information Forensics and Security*, no. 99, 2017.

[7] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, "Image analogies," in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '01. New York, NY, USA: ACM, 2001, pp. 327–340. [Online]. Available: http://doi.acm.org/10.1145/383259.383295

[8] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *CoRR*, vol. abs/1611.07004, 2016. [Online]. Available: http://arxiv.org/abs/1611.07004

[9] P. Sangkloy, J. Lu, C. Fang, F. Yu, and J. Hays, "Scribbler: Controlling deep image synthesis with sketch and color," *CoRR*, vol. abs/1612.00835, 2016. [Online]. Available: http://arxiv.org/abs/1612.00835

[10] J. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *CoRR*, vol. abs/1703.10593, 2017. [Online]. Available: http://arxiv.org/abs/1703.10593

[11] I. Melekhov, J. Kannala, and E. Rahtu, "Siamese network features for image matching," in *2016 23rd International Conference on Pattern Recognition (ICPR)*, Dec 2016, pp. 378–383.

Curriculum Vitae

Name: Raghav Sethi

Date of birth: 30/05/1996

Education qualifications:

- **Indian Institute of Technology Mandi** 2014 - present
  B.Tech Computer Science & Engineering *CGPA: 8.06*
- **Mother Khazani Convent School, New Delhi** 2014
  Class XII CBSE, School Topper *Percentage: 97 %*
- **Jankidas Kapur Public School, Sonipat** CGPA: 10
  Class X CBSE *2011 - 2012*

Permanent address: H.No 1315 Sector-14 Sonipat, Haryana